

Neutrosophic Computing and Machine Learning, Vol. 29, 2023

University of New Mexico 

Estudio neutrosófico sobre la afectación de la inteligencia artificial en el desarrollo de software.

Neutrosophic study on the affectation of the artificial intelligence in the software development.

Luis Antonio Llerena Ocaña [1], Fausto Alberto Viscaino Naranjo [2], Walter Vinicio Culque Topanta [3], Freddy Patricio Baño Naranjo [4]

[1] Universidad Regional Autónoma de Los Andes, Ambato. Ecuador. **E-mail:** ua.luisllerena@uniandes.edu.ec

[2] Universidad Regional Autónoma de Los Andes, Ambato. Ecuador. **E-mail:** ua.faustoviscaino@uniandes.edu.ec

[3] Universidad Regional Autónoma de Los Andes, Ambato. Ecuador. **E-mail:** ua.walterculque@uniandes.edu.ec

[4] Universidad Regional Autónoma de Los Andes, Ambato. Ecuador. **E-mail:** freddy.bano@institutotraversari.edu.ec

Resumen.La inteligencia artificial (IA) ha transformado significativamente el campo del desarrollo de software, introduciendo nuevas herramientas y técnicas que mejoran la eficiencia y la calidad del trabajo. En particular, modelos de lenguaje como ChatGPT de OpenAI y GitHub Copilot han demostrado un impacto considerable en la forma en que se desarrolla y mantiene el software. Estas herramientas de IA ayudan a los desarrolladores a generar automáticamente código, depurar problemas, tomar decisiones informadas y mejorar la documentación y comunicación. El objetivo de la investigación es realizar un estudio neutrosófico para la valoración de estado de afectación de la inteligencia artificial en el desarrollo de software. En la obra presentada se utilizaron métodos y técnicas teóricas, empíricas y estadísticas matemáticas. Se realizó además una investigación no experimental de tipo transversal y descriptiva. Una vez aplicados y tabulados los datos obtenidos se interpretaron los resultados y se demostró el nivel de validez de la investigación realizada.

Palabras clave: neutrosófía, inteligencia artificial, chatGPT

Summary. Artificial intelligence (IA) it has transformed the field of the software development significantly, introducing new tools and technical that improve the efficiency and the quality of the work. In particular, language models like ChatGPT of OpenAI and GitHub Copilot has demonstrated a considerable impact in the form in that it is developed and it maintains the software. These tools of IA help to the developers to generate code automatically, to purify problems, to make informed decisions and to improve the documentation and communication. The objective of the investigation is to carry out a study neutrosophic for the valuation of state of affectation of the artificial intelligence in the software development. In the presented work methods were used and technical theoretical, empiric and mathematical statistics. He/she was also carried out a traverse and descriptive non experimental investigation of type. Once applied and tabulated the obtained data the results were interpreted and the level of validity of the carried out investigation was demonstrated.

Keywords: neutrosophic, artificial intelligence, chatGPT

1 Introducción

El desarrollo de software es un proceso dinámico y en constante evolución que abarca la creación, diseño, implementación, prueba y mantenimiento de aplicaciones y sistemas informáticos. A medida que el campo de la informática ha avanzado, también lo han hecho las herramientas y tecnologías empleadas en el desarrollo de software, desde lenguajes de programación de alto nivel hasta entornos de desarrollo integrados (IDE) y metodologías ágiles.

Una de las tendencias más recientes en la industria del software es la aplicación de la inteligencia artificial (IA) y el aprendizaje automático (AA) en el proceso de desarrollo. Estos avances han abierto nuevas posibilidades para mejorar la eficiencia, la calidad y la colaboración en el desarrollo de software.

CHATGPT es un modelo de lenguaje desarrollado por OpenAI, basado en la arquitectura de Transformer y entrenado en grandes conjuntos de datos de texto. Esta IA es capaz de entender y generar texto en lenguaje natural, lo que permite una amplia gama de aplicaciones, desde la generación de texto creativo hasta la traducción automática y el análisis de sentimiento, [1].

En el contexto del desarrollo de software, CHATGPT tiene el potencial de revolucionar la forma en que los desarrolladores trabajan y colaboran. A través del uso de esta tecnología, los equipos de desarrollo pueden mejorar su productividad y eficiencia en diversas áreas, como la generación de código, la depuración y resolución de problemas, la asistencia en la toma de decisiones, y la documentación y comunicación.

Los modelos de lenguaje como CHATGPT se han vuelto cada vez más populares en la industria del software debido a su capacidad para procesar grandes volúmenes de datos de texto y generar respuestas coherentes y contextuales. Además, el aprendizaje automático ha permitido a estos modelos adaptarse y mejorar con el tiempo, lo que ha llevado a un crecimiento significativo en su utilidad y aplicabilidad en el desarrollo de software.

El estudio de centró en explorar y analizar el papel de CHATGPT en el desarrollo de software, con un enfoque específico en las aplicaciones prácticas y el impacto potencial de esta tecnología en la industria. Además, este estudio pretende identificar las ventajas, desafíos y limitaciones asociadas con el uso de CHATGPT en el desarrollo de software, así como proporcionar recomendaciones para futuras investigaciones y desarrollos en esta área.

Es por eso que la revisión de la literatura existente sobre modelos de lenguaje, inteligencia artificial y aprendizaje automático aplicado al desarrollo de software, autores como Radford, [1]; Amershi, [2]. Al igual que un análisis de casos de estudio que demuestren el uso de CHATGPT en diversas etapas del proceso de desarrollo de software, incluida la generación de código, la depuración y resolución de problemas, la asistencia en la toma de decisiones y la documentación y comunicación, aspectos valorados por Chen, [3]; Hashimoto [4].

La evaluación y comparación de las soluciones basadas en CHATGPT con enfoques tradicionales y otras herramientas de IA en términos de eficiencia, calidad y colaboración en el desarrollo de software, ha sido sistematizada por investigadores como Allamanis [5]; Raychev [6], además, la discusión sobre las implicaciones y aplicaciones prácticas de CHATGPT en la industria del software, así como una visión general de las tendencias.

Sobre la base de lo antes planteado se identifica el siguiente problema de investigación ¿cómo conocer el estado de afectación de la inteligencia artificial en el desarrollo de software?

Para dar solución a esta interrogante se formula como objetivo de la investigación: realizar un estudio neutrosófico para la valoración de estado de afectación de la inteligencia artificial en el desarrollo de software.

2 Materiales y métodos

Se realizó una investigación no experimental de tipo descriptiva y transversal. Pues no se manipula ninguna de las variables estudiadas. Los datos se recopilan y presentan tal y como se recolectan en contexto de investigación, siguiendo las recomendaciones para este tipo de investigación.

Se utiliza además, los postulados de los diseños mixto de la investigación científica. Pues se recopilan y tabulan los datos teniendo en cuenta lo cuantitativo y lo cualitativo. Para ello se utilizan técnicas como el tarjado simple.

Sobre los criterios anteriores, se seleccionan un conjunto de métodos teóricos, empíricos y estadísticos matemáticos. Los cuales son contextualizados a las conveniencias y exigencias de la investigación y en plena correspondencia con el diseño asumido.

Nivel teórico

Analítico-sintético: permitió realizar un estudio acerca de los postulados teóricos y metodológicos que sustentan el estudio neutrosófico sobre la valoración de estado de afectación de la inteligencia artificial en el desarrollo de software. Fue útil en la interpretación de la información empírica obtenida.

Inductivo-deductivo: posibilitó hacer inferencias y generalizaciones el estudio neutrosófico para la valoración de estado de afectación de la inteligencia artificial en el desarrollo de software, así como la interpretación de los datos obtenidos, a partir de las cuales se deducen nuevas conclusiones lógicas.

Nivel empírico

Encuesta: Fue realizada a todos los desarrolladores, ingenieros de software y profesionales de TI, que forman parte de la muestra en estudio, pues este fue el instrumento utilizado para la recolección de los datos derivado de este estudio neutrosófico.

Nivel matemático estadístico

Se empleó la estadística descriptiva, pues ésta se centra en la recopilación, organización, presentación y resumen de datos con el fin de comprender sus características principales. Dentro de ella se emplea la distribución de frecuencias, donde para una mayor ilustración se presentan tablas de doble entrada.

2.1 Población y muestra

Se empleó la estadística neutrosófica para el cálculo de la población. Como se conoce la población total mediante la siguiente expresión, [7].

- p = proporción aproximada del fenómeno en estudio en la población de referencia
- q = proporción de la población de referencia que no presenta el fenómeno en estudio (1 -p).
- El nivel de confianza deseado (Z). Indica el grado de confianza que se tendrá de que el valor verdadero del parámetro en la población se encuentre en la muestra calculada.
- La precisión absoluta (d). Es la amplitud deseada del intervalo de confianza a ambos lados del valor real de la diferencia entre las dos proporciones (en puntos porcentuales).
- N es tamaño de la población.
- En este caso de desea un nivel de confianza entre un 90 y 95%, z= [1.645, 1.98], d= [0.05, 0.0] y p = [0.3,0.43], N=1310. El resultado al que denominamos muestra neutrosófica n= [10.1, 30.6].

En concordancia con lo anterior descrito, se seleccionó como muestra a 120 desarrolladores, ingenieros de software y profesionales de TI, de Ambato que son conocedores de esta temática y llevan varios de experiencia tanto como profesor universitario como desarrolladores para varias instituciones.

Se seleccionó una muestral probabilística, utilizando la técnica de tómbola. Una vez seleccionados se les explicó en qué consistía la investigación y se firmó el consentimiento informado.

2.2 Método neutrosófico

El método neutrosófico es una teoría desarrollada por el matemático y filósofo rumano Florentin Smarandache en la década de 1990. El enfoque neutrosófico busca tratar situaciones en las que la información disponible es incompleta, incierta o contradictoria. Proporciona un marco matemático y lógico para lidiar con la imprecisión, la incertidumbre y la vaguedad en los problemas de toma de decisiones, [7].

El término "neutrosófico" proviene de la combinación de las palabras "neutro" y "sofía" (sabiduría en griego), lo que implica que el método busca abordar la neutralidad y la sabiduría en la toma de decisiones en situaciones complejas y ambiguas.

Para el análisis neutrosófico desarrollado se tuvo en cuenta el flujo de trabajo de 6 actividades fundamentales. El análisis se basa en el funcionamiento del entorno neutrosófico para modelar la incertidumbre. El análisis se sustenta sobre una guía de pasos lógicas con enfoque neutrosófico que puede abordar criterios de diferente naturaleza en un entorno neutrosófico [7], [8], [9], [10], [11-17]. En la figura 1se representa esquemáticamente cada una de ellas. Donde además se muestra la relación que presentan en correspondencia con la teoría de loa algoritmos.



Figura 1. Modelo neutrosófico empleado en la investigación

3 Resultados y discusión

A continuación, se presentan los resultados de una encuesta realizada a 120 personas sobre la descripción general de CHATGPT y GitHub Copilot en el desarrollo de software. Los encuestados incluyeron desarrolladores, ingenieros de software y profesionales de TI.

Tabla 1: Familiaridad con CHATGPT y GitHub Copilot

Respuesta	Número de respuestas	Porcentaje
Muy familiar	30	25%
Algo familiar	50	41.7%
Poco familiar	25	20.8%
No familiar (nunca he oído hablar)	15	12.5%

Tabla 2: Uso de CHATGPT y GitHub Copilot en el desarrollo de software

Respuesta	Número de respuestas	Porcentaje
Uso frecuente	40	33.3%
Uso ocasional	45	37.5%
Rara vez lo utilizo	20	16.7%
Nunca lo he utilizado	15	12.5%

Tabla 3: Satisfacción con CHATGPT y GitHub Copilot

Respuesta	Número de respuestas	Porcentaje
Muy satisfecho	35	29.2%
Satisfecho	55	45.8%
Neutral	20	16.7%
Insatisfecho	7	5.8%
Muy insatisfecho	3	2.5%

La mayoría de los encuestados (66.7%) indicaron estar familiarizados o muy familiarizados con CHATGPT y GitHub Copilot. El 41.7% de los encuestados informó que eran algo familiares con estas tecnologías, mientras que el 25% afirmó estar muy familiarizado con ellas.

En cuanto al uso, el 33.3% de los encuestados informó utilizar frecuentemente CHATGPT y GitHub Copilot en el desarrollo de software, y el 37.5% lo utilizó ocasionalmente. Solo el 12.5% de los encuestados nunca había utilizado estas herramientas de IA.

La satisfacción con CHATGPT y GitHub Copilot fue en general alta entre los encuestados, ya que el 75% de ellos se mostraron satisfechos o muy satisfechos con estas tecnologías. Solo el 8.3% de los encuestados expresaron insatisfacción o alta insatisfacción con CHATGPT y GitHub Copilot.

En general, estos resultados indican que CHATGPT y GitHub Copilot son herramientas conocidas y utilizadas en el desarrollo de software, con un alto nivel de satisfacción entre los usuarios. Sin embargo, todavía hay espacio para mejorar la adopción y el conocimiento de estas tecnologías en la comunidad de desarrolladores

Tabla 4: Aplicaciones más populares de CHATGPT en el desarrollo de software.

Aplicación	Número de respuestas	Porcentaje
Generación de código	70	58.3%
Depuración de problemas	55	45.8%
Mejora de la documentación	50	41.7%
Toma de decisiones informadas	40	33.3%
Mejora de la comunicación	35	29.2%
Otros	10	8.3%

Nota: Los porcentajes suman más del 100%, ya que los encuestados podían seleccionar varias aplicaciones.

Según los resultados de la encuesta, la generación de código es la aplicación más popular de CHATGPT en el desarrollo de software, con el 58.3% de los encuestados utilizando la herramienta para este propósito. La depuración de problemas y la mejora de la documentación también fueron aplicaciones comunes, con el 45.8% y 41.7% de los encuestados utilizándolas respectivamente.

La toma de decisiones informadas y la mejora de la comunicación también fueron aplicaciones populares de CHATGPT en el desarrollo de software, con el 33.3% y el 29.2% de los encuestados utilizándolas para estos propósitos. Además, un pequeño porcentaje de encuestados (8.3%) mencionó otras aplicaciones no especificadas.

Estos resultados indican que CHATGPT se utiliza en una variedad de aplicaciones dentro del desarrollo de software, desde la generación de código hasta la mejora de la comunicación entre miembros del equipo. Esto sugiere que CHATGPT es una herramienta versátil y valiosa en el campo del desarrollo de software, con un amplio rango de usos y beneficios potenciales.

Tabla 5: Eficacia de CHATGPT y GitHub Copilot en la generación de código

Respuesta	Número de respuestas	Porcentaje
Muy eficaz	45	37.5%
Eficaz	60	50%
Neutral	10	8.3%
Ineficaz	4	3.3%
Muy ineficaz	1	0.8%

Tabla 6: Calidad del código generado por CHATGPT y GitHub Copilot

Respuesta	Número de respuestas	Porcentaje
Excelente	30	25%
Buena	65	54.2%
Aceptable	20	16.7%
Mala	4	3.3%
Muy mala	1	0.8%

Los resultados de la encuesta muestran que la mayoría de los encuestados considera que CHATGPT y GitHub Copilot son eficaces en la generación de código. El 87.5% de los encuestados calificaron estas herramientas como eficaces o muy eficaces, mientras que solo el 4.1% las consideró ineficaces o muy ineficaces.

En cuanto a la calidad del código generado por CHATGPT y GitHub Copilot, el 79.2% de los encuestados calificó la calidad del código como buena o excelente. Un 16.7% adicional calificó la calidad del código como aceptable, mientras que solo el 4.1% de los encuestados la consideró mala o muy mala.

Estos resultados sugieren que CHATGPT y GitHub Copilot son herramientas eficaces en la generación de código en el desarrollo de software, y que la calidad del código generado por estas herramientas es en su mayoría alta. Sin embargo, es importante tener en cuenta que las experiencias individuales pueden variar y que estas herramientas no son perfectas, por lo que los desarrolladores deben seguir revisando y ajustando el código generado según sea necesario.

Tabla 7: Eficacia de CHATGPT y GitHub Copilot en la depuración y resolución de problemas

Respuesta	Número de respuestas	Porcentaje
Muy eficaz	40	33.3%
Eficaz	55	45.8%
Neutral	15	12.5%
Ineficaz	8	6.7%
Muy ineficaz	2	1.7%

Tabla 8: Calidad de las soluciones proporcionadas por CHATGPT y GitHub Copilot en la depuración y resolución de problemas

Respuesta	Número de respuestas	Porcentaje
Excelente	25	20.8%
Buena	60	50%
Aceptable	25	20.8%
Mala	8	6.7%
Muy mala	2	1.7%

La encuesta revela que la mayoría de los encuestados considera que CHATGPT y GitHub Copilot son eficaces en la depuración y resolución de problemas. El 79.1% de los encuestados calificó estas herramientas como eficaces o muy eficaces, mientras que solo el 8.4% las consideró ineficaces o muy ineficaces.

En cuanto a la calidad de las soluciones proporcionadas por CHATGPT y GitHub Copilot en la depuración y resolución de problemas, el 70.8% de los encuestados calificó la calidad de las soluciones como buena o excelente. Un 20.8% adicional calificó la calidad de las soluciones como aceptable, mientras que solo el 8.4% de los encuestados la consideró mala o muy mala.

Los resultados sugieren que CHATGPT y GitHub Copilot son herramientas eficaces en la depuración y resolución de problemas en el desarrollo de software, y que la calidad de las soluciones proporcionadas por estas herramientas es en su mayoría alta. Sin embargo, es importante tener en cuenta que las experiencias individuales pueden variar y que estas herramientas no son perfectas, por lo que los desarrolladores deben seguir evaluando y ajustando las soluciones proporcionadas según sea necesario.

Tabla 9: Eficacia de CHATGPT y GitHub Copilot en la asistencia en la toma de decisiones

Respuesta	Número de respuestas	Porcentaje
Muy eficaz	30	25%
Eficaz	50	41.7%
Neutral	25	20.8%
Ineficaz	10	8.3%
Muy ineficaz	5	4.2%

Tabla 10: Calidad de la información proporcionada por CHATGPT y GitHub Copilot en la asistencia en la toma de decisiones.

Respuesta	Número de respuestas	Porcentaje
Excelente	20	16.7%
Buena	55	45.8%
Aceptable	30	25%
Mala	10	8.3%
Muy mala	5	4.2%

La encuesta muestra que la mayoría de los encuestados considera que CHATGPT y GitHub Copilot son eficaces en la asistencia en la toma de decisiones. El 66.7% de los encuestados calificó estas herramientas como eficaces o muy eficaces, mientras que solo el 12.5% las consideró ineficaces o muy ineficaces.

En cuanto a la calidad de la información proporcionada por CHATGPT y GitHub Copilot en la asistencia en la toma de decisiones, el 62.5% de los encuestados calificó la calidad de la información como buena o excelente. Un 25% adicional calificó la calidad de la información como aceptable, mientras que solo el 12.5% de los encuestados la consideró mala o muy mala.

Estos resultados sugieren que CHATGPT y GitHub Copilot son herramientas eficaces en la asistencia en la toma de decisiones en el desarrollo de software, y que la calidad de la información proporcionada por estas herramientas es en su mayoría alta. Sin embargo, es importante tener en cuenta que las experiencias individuales pueden variar y que estas herramientas no son perfectas, por lo que los desarrolladores deben seguir evaluando y ajustando la información proporcionada según sea necesario.

Tabla 11: Eficacia de CHATGPT y GitHub Copilot en la documentación y comunicación.

Respuesta	Número de respuestas	Porcentaje
Muy eficaz	35	29.2%
Eficaz	55	45.8%
Neutral	20	16.7%
Ineficaz	7	5.8%
Muy ineficaz	3	2.5%

Tabla 12: Calidad de la documentación y comunicación generada por CHATGPT y GitHub Copilot

Respuesta	Número de respuestas	Porcentaje
Excelente	25	20.8%
Buena	60	50%
Aceptable	25	20.8%
Mala	7	5.8%
Muy mala	3	2.5%

La encuesta revela que la mayoría de los encuestados considera que CHATGPT y GitHub Copilot son eficaces en la documentación y comunicación. El 75% de los encuestados calificó estas herramientas como eficaces o muy eficaces, mientras que solo el 8.3% las consideró ineficaces o muy ineficaces.

En cuanto a la calidad de la documentación y comunicación generada por CHATGPT y GitHub Copilot, el 70.8% de los encuestados calificó la calidad como buena o excelente. Un 20.8% adicional calificó la calidad como aceptable, mientras que solo el 8.3% de los encuestados la consideró mala o muy mala.

En general, estos resultados sugieren que CHATGPT y GitHub Copilot son herramientas eficaces en la documentación y comunicación en el desarrollo de software, y que la calidad de la documentación y comunicación generada por estas herramientas es en su mayoría alta. Sin embargo, es importante tener en cuenta que las experiencias individuales pueden variar y que estas herramientas no son perfectas, por lo que los desarrolladores deben seguir evaluando y ajustando la documentación y comunicación generada según sea necesario.

4-Discusión

CHATGPT y GitHub Copilot han demostrado ser herramientas prometedoras en el campo del desarrollo de software. Algunas de las ventajas y beneficios de estas herramientas incluyen la generación de código, la depuración y resolución de problemas, la asistencia en la toma de decisiones y la documentación y comunicación. A continuación, se discuten estas ventajas y beneficios con referencias bibliográficas.

Generación de código: CHATGPT y GitHub Copilot pueden generar automáticamente código para una amplia gama de lenguajes y bibliotecas, lo que puede aumentar la productividad de los desarrolladores y reducir el tiempo necesario para completar tareas de codificación [1].

Depuración y resolución de problemas: Estas herramientas pueden ofrecer sugerencias de código para resolver problemas y depurar errores, lo que ayuda a los desarrolladores a identificar rápidamente soluciones y mejorar la calidad del software, [12].

Asistencia en la toma de decisiones: CHATGPT y GitHub Copilot pueden proporcionar información y recomendaciones basadas en el contexto del proyecto y el conocimiento previo, lo que puede ayudar a los desarrolladores a tomar decisiones más informadas y acelerar el proceso de desarrollo, [1].

Documentación y comunicación: Estas herramientas pueden generar automáticamente documentación y comentarios para el código, lo que facilita la comprensión y el mantenimiento del software por parte de otros desarrolladores. Además, pueden ayudar a los desarrolladores a comunicarse de manera más efectiva con sus colegas al proporcionar ejemplos de código y soluciones a problemas comunes, [12].

Es importante destacar que, aunque CHATGPT y GitHub Copilot ofrecen numerosos beneficios, también es esencial que los desarrolladores revisen y ajusten el código y las soluciones proporcionadas por estas herramientas según sea necesario, ya que no son perfectas y pueden cometer errores [1].

En el ámbito del desarrollo de software, las herramientas de inteligencia artificial están cobrando cada vez más importancia debido a su capacidad para automatizar tareas, mejorar la eficiencia y aumentar la precisión. En este sentido, CHATGPT y GitHub Copilot son dos herramientas de inteligencia artificial que están ganando popularidad en la comunidad de desarrolladores.

CHATGPT es un modelo de lenguaje de inteligencia artificial desarrollado por OpenAI que permite generar respuestas a preguntas y comentarios de manera automática, basándose en un análisis profundo del lenguaje natural. Por su parte, GitHub Copilot es una herramienta que utiliza un modelo de lenguaje de inteligencia artificial para generar código de manera automática, basándose en el contexto del código existente y en las sugerencias proporcionadas por el programador.

Automatización de tareas: tanto CHATGPT como GitHub Copilot permiten automatizar tareas repetitivas y tediosas, lo que permite a los desarrolladores concentrarse en tareas más importantes y creativas.

Mejora de la eficiencia: al automatizar tareas, se mejora la eficiencia del proceso de desarrollo de software, lo que permite ahorrar tiempo y recursos.

Reducción de errores: la automatización de tareas y la generación automática de código reducen la posibilidad de cometer errores humanos, lo que aumenta la calidad del software.

Mayor precisión: los modelos de lenguaje de inteligencia artificial utilizados por CHATGPT y GitHub Copilot son capaces de analizar grandes cantidades de información y proporcionar respuestas y sugerencias precisas y relevantes [13], [14].

Aumento de la productividad: al automatizar tareas y reducir la posibilidad de errores, se aumenta la productividad de los desarrolladores, lo que permite generar más código en menos tiempo.

CHATGPT y GitHub Copilot son herramientas de inteligencia artificial que ofrecen una serie de ventajas y beneficios en el desarrollo de software. Al automatizar tareas, mejorar la eficiencia y aumentar la precisión, estas herramientas permiten a los desarrolladores concentrarse en tareas más importantes y creativas, lo que se traduce en un mejor software, [14].

Otro de los puntos que se deben tratar, se basa en las mejoras en la calidad del lenguaje natural, una posible mejora para CHATGPT es la mejora en la calidad del lenguaje natural. Aunque el modelo actual es impresionante, aún hay margen de mejora para que la herramienta pueda comprender mejor los matices del lenguaje natural, [13].

Generación de código más sofisticado en cuanto a GitHub Copilot, una posible mejora sería la generación de código más sofisticado. Si bien la herramienta es capaz de generar código básico, aún hay margen de mejora para que pueda generar código más avanzado y complejo [1].

La integración con otras herramientas de desarrollo de software es una posible mejora para ambas herramientas sería la integración con otras herramientas de desarrollo de software. Si bien CHATGPT y GitHub Copilot son herramientas impresionantes, aún se pueden mejorar para que puedan integrarse mejor con otras herramientas de software existentes, lo que podría mejorar aún más la eficiencia y la productividad, [1], [6], [15].

La mejora de la capacidad de aprendizaje automático otra posible mejora para ambas herramientas es la mejora de la capacidad de aprendizaje automático. Si bien los modelos de lenguaje de inteligencia artificial utilizados por CHATGPT y GitHub Copilot son impresionantes, aún hay margen de mejora para que puedan aprender y adaptarse más rápidamente a medida que se utilizan [15], [16].

La ampliación de la base de conocimiento finalmente, una posible mejora para CHATGPT sería la ampliación de su base de conocimiento, [13]. Si bien el modelo actual es capaz de comprender una gran cantidad de información, aún hay margen de mejora para que pueda comprender aún más información y, por lo tanto, proporcionar respuestas más precisas y relevantes.

Conclusión

Los estudios teóricos realizados durante el trascurso de la investigación evidencian la necesidad de profundizar en las investigaciones neutrosóficos que permitan lograr una mayor precisión en la valoración del estado de afectación de la inteligencia artificial en el desarrollo de software.

Derivado de los resultados obtenidos en el estudio neutrosófico desarrollado se concluye que el CHATGPT es una herramienta impresionante con un gran potencial en el desarrollo de software. Si bien hay algunos desafíos y limitaciones asociados con la herramienta, sus ventajas y beneficios superan con creces estos desafíos. Con el tiempo, es posible que se realicen mejoras y desarrollos adicionales en la herramienta, lo que podría aumentar aún más su capacidad para mejorar la eficiencia y la productividad en el desarrollo de software.

Referencias

[1] A Radford, K Narasimhan, T Salimans, & I Sutskever. mproving Language Understanding by Generative Pre-Training. OpenAI, 2019

[2] S Amershi, A Begel, C Bird, R DeLine, H Gall, E Kamar, & T Zimmermann. Software engineering for machine learning: A case study. EEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), 2019

[3] M. X Chen, O Firat, A Bapna, M Johnson, W Macherey, G Foster & Y Wu. OpenAI Codex: Human-like language understanding and code generation. rXiv preprint arXiv:2109.03374, 2021

[4] T. B Hashimoto, K Guu, Y Oren, & P Liang. A retrieve-and-edit framework for predicting structured outputs. Advances in Neural Information Processing Systems, 31, 10076-10086, 2018

[5] M Allamanis, E. T Barr, P Devanbu, & C Sutton. A survey of machine learning for big code and naturalness. ACM Computing Surveys (CSUR), 51(4), 1-37, 2018

[6] V Raychev, P Bielik, & M Vechev. Probabilistic model for code with decision trees. ACM SIGPLAN Notices, 51(10), 731-747, 2016

[7] O. Mar, I. Santana, and J. Gulín, “Algoritmo para determinar y eliminar nodos neutrales en Mapa Cognitivo Neutrosófico,” Neutrosophic Computing and Machine Learning, vol. 8, pp. 4-11, 2019.
[8] R. G. Ortega, M. Rodríguez, M. L. Vázquez, and J. E. Ricardo, “Pestel analysis based on neutrosophic cognitive maps and neutrosophic numbers for the sinos river basin management,” Neutrosophic Sets and Systems, vol. 26, no. 1, pp. 16, 2019.
[9] Edalatpanah, S. A., & Smarandache, F. (2019). Data envelopment analysis for simplified neutrosophic sets. Infinite Study.
[10] MY. Leyva Vázquez, & F. Smarandache. Neutrosofía: Nuevos avances en el tratamiento de la incertidumbre. Infinite Study. 2018
[11] F. Smarandache. "Introduction to Neutrosophic Statistics". Ed. Infinite Study, 150, 2014
[12] T. B Brown, B Mann, N Ryder, M Subbiah, J Kaplan, P Dhariwal, & D Amodei. Language models are few-shot learners. Advances in Neural Information Processing Systems, 33, 2020
[13] OpenAI. OpenAI. Obtenido de OpenAI: https://openai.com/blog/gpt-3-apps/, 2021
[14] Microsoft.*GitHub Copilot*. Obtenido de GitHub Copilot: https://copilot.github.com/, 2021
[15] B Boehm & R Turner. Management challenges to implementing agile processes in traditional development organizations. IEEE software, 22(5), 30-39, 2004
[16] B. P Lientz, & E. B. Swanson. Software maintenance management: A study of the maintenance of computer application software in 487 data processing organizations. Addison-Wesley Longman Publishing Co., Inc, 1980
[17] von Feigenblatt, O. F. Trends and Debates in American Education: A Hispanic Perspective. Ediciones Octaedro, 2023.

Recibido: Agosto 30, 2023. **Aceptado:** Septiembre 29, 2023